



PL/SQL Cursor

Disciplina

Database Application Development

Leonardo Uchida

Referências Bibliográficas

Manuais ou Materiais oficiais da Oracle ou Oracle Press

*Oracle9i / 10g PL/SQL * Guia de Consulta Rápida – Celso Henrique Poderoso*

Este material é apenas um guia de estudo e não substitui a leitura da referência bibliográfica e a consulta de anotações de sala de aula

Objetivos

Depois de completar esta lição, você poderá fazer o seguinte:

- Criar uma instrução SELECT em PL/SQL
- Declarar o tipo de dados e tamanho de uma variável PL/SQL dinamicamente
- Criar instruções SELECT em PL/SQL
- Controlar transações em PL/SQL
- Determinar o resultado das instruções DML SQL

Instruções SQL em PL/SQL

- Extrair uma linha de dados de um banco de dados usando o comando **SELECT**.
Apenas um único conjunto de valores pode ser retornado.
- Fazer alterações nas linhas no banco de dados usando os comandos **DML**.
- Controlar uma transação com o comando **COMMIT**, **ROLLBACK** ou **SAVEPOINT**.
- Determinar o resultado do **DML** com cursores implícitos.

Instruções SELECT em PL/SQL

Recuperar dados do banco de dados com SELECT.

Sintaxe

```
SELECT  select_list
INTO    {variable_name[, variable_name]...
        | record_name}
FROM    tabela
WHERE   condição;
```

Instruções SELECT em PL/SQL

A cláusula INTO é necessária.

Exemplo

```
DECLARE
  v_deptno  NUMBER(2);
  v_loc     VARCHAR2(15);
BEGIN
  SELECT    deptno, loc
  INTO      v_deptno, v_loc
  FROM      dept
  WHERE     dname = 'SALES';
  ...
END;
```

Recuperando Dados em PL/SQL

Recuperar a data da ordem de compra e de entrega para a ordem especificada.

Exemplo

```
DECLARE
  v_orderdate  ord.orderdate%TYPE;
  v_shipdate   ord.shipdate%TYPE;
BEGIN
  SELECT  orderdate, shipdate
  INTO    v_orderdate, v_shipdate
  FROM    ord
  WHERE   id = 620;
  ...
END;
```

- *As exceções NO_DATA_FOUND, TOO_MANY_ROWS e OTHERS são as mais usadas para esse tipo de acesso ao banco de dados*

Recuperando Dados em PL/SQL

Retornar o total de salários de todos os funcionários no departamento especificado.

Exemplo

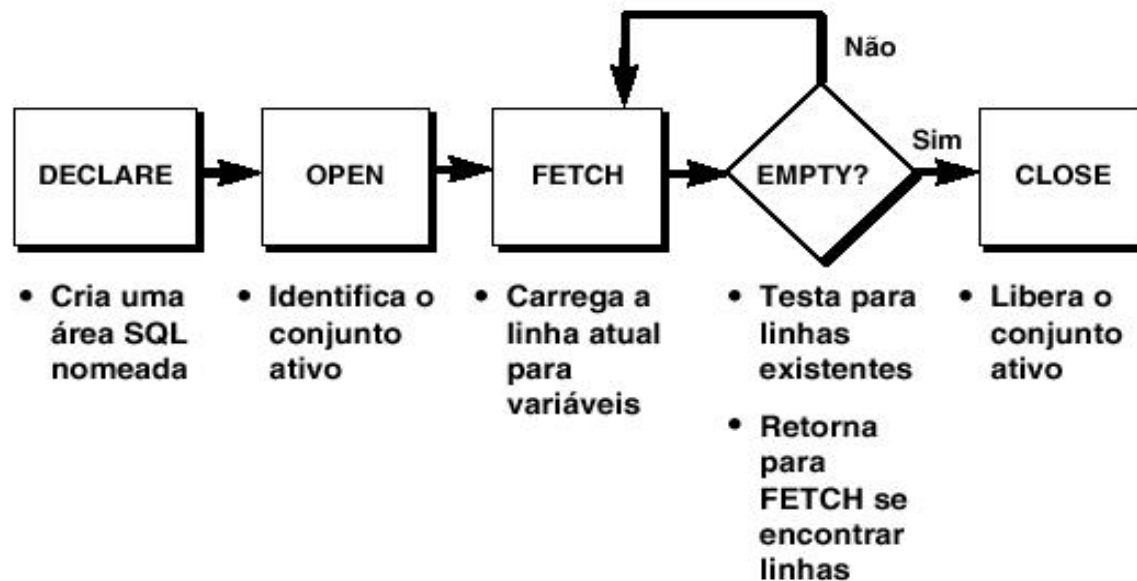
```
DECLARE
  v_sum_sal    emp.sal%TYPE;
  v_deptno     NUMBER NOT NULL := 10;
BEGIN
  SELECT      SUM(sal) -- group function
  INTO        v_sum_sal
  FROM        emp
  WHERE       deptno = v_deptno;
END;
```

- Como tratar um conjunto de linhas (0 ou n linhas) retornadas em um *SELECT* dentro de um bloco PL/SQL.
- Por que existe Cursor ?
 - quando necessita-se utilizar em um bloco PL/SQL um *SELECT* que retorna:
 - » várias linhas: `TOO_MANY_ROWS`
 - » nenhuma linha: `NO_DATA_FOUND`
- Quando selecionamos várias linhas no SQL*Plus ou SQL*Developer, essas ferramentas utilizam internamente um `CURSOR` para gerenciar as informações exibidas.

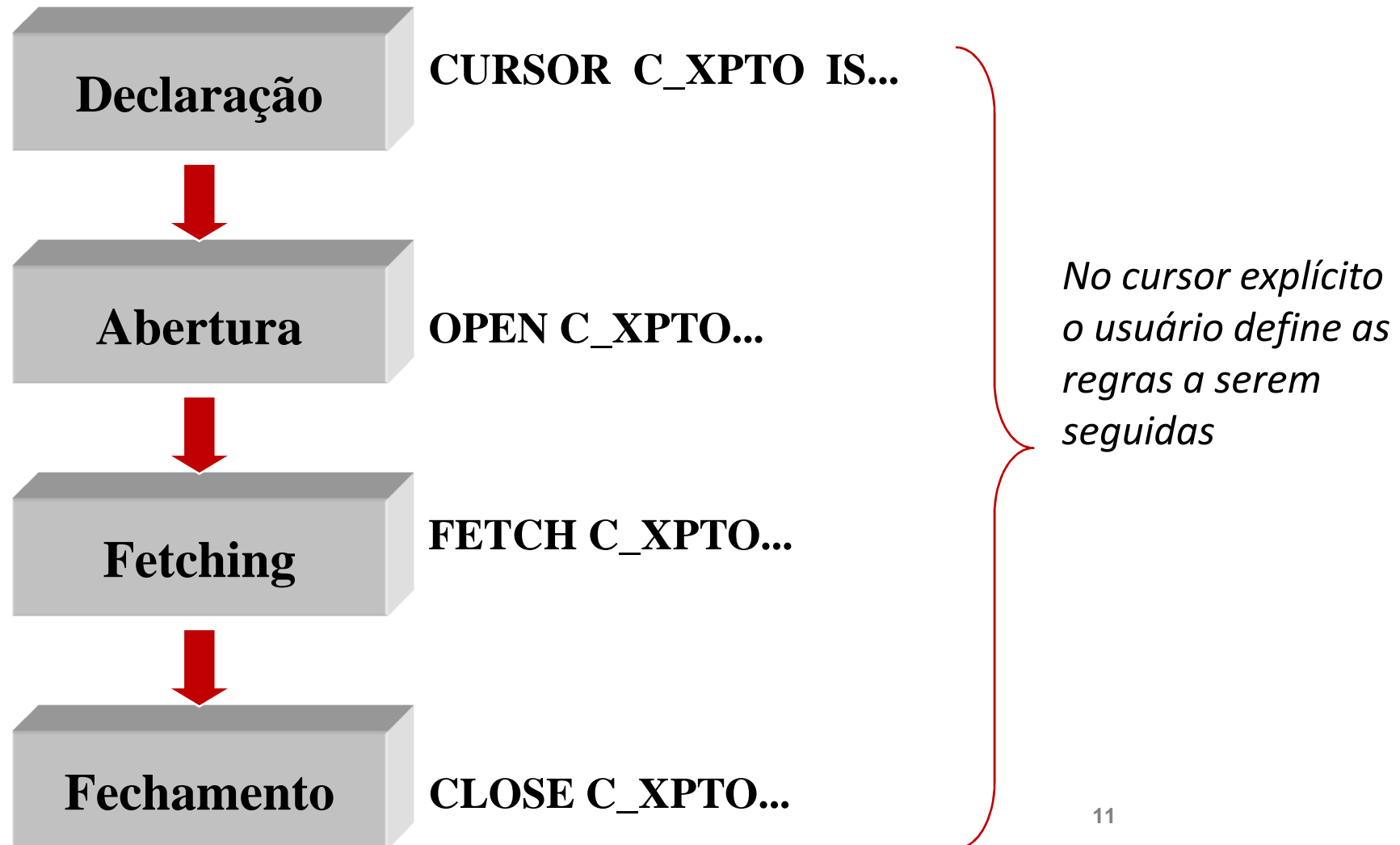
- O que é :
 - São áreas de memória compostas de linhas e colunas que servem para armazenar o resultado de uma seleção, que retorna 0 ou mais linhas

- Existem dois tipos:
 - Explícitos (você declara, abre e fecha)
 - Implícitos (o PL/SQL trata de abrir e fechá-lo, espera uma única linha)

Controlando Cursores Explícitos



Etapas de um cursor explícito



Declarando um cursor explícito

- *Sintaxe:*
*CURSOR nome [(parâmetro,...)] [RETURN tipo_dado] IS
SELECT ...*
- *Utilizado na cláusula DECLARE do PL/SQL*
- *Limitado apenas pela memória*

Declarando um cursor explícito

```
DECLARE  
CURSOR C_CLIENTE IS  
SELECT CD_CLIENTE, NM_CLIENTE FROM LOC_CLIENTE;
```

```
v_cliente c_cliente%rowtype; -- colunas do cursor
```

```
BEGIN  
    NULL;  
END;
```

Abertura de um cursor explícito

- *Sintaxe:*
OPEN [cursor];
- *Deve ser utilizado após a cláusula BEGIN do bloco PL/SQL*
- *Fica posicionado na primeira linha*
- *Atributos de um cursor*
 - *%ISOPEN , %NOTFOUND,*
 - *%FOUND e %ROWCOUNT*

Abertura de um cursor explícito

```
DECLARE  
CURSOR C_CLIENTE IS  
SELECT CD_CLIENTE COD, NM_CLIENTE NOME_COMPLETO FROM  
LOC_CLIENTE;
```

```
v_cliente c_cliente%rowtype; -- colunas do cursor
```

```
BEGIN  
    OPEN C_CLIENTE;  
END;
```

Exemplo: Abertura de um cursor explícito

Este comando gera na memória uma tabela semelhante a exibida abaixo, com o primeiro ponteiro apontando para a primeira linha

Linha Corrente

COD	NOME_COMPLETO
1	José da Silva
2	Karina Souza
3	Klaus Mimi
4

- *Acessa as linhas do Cursor após abertas*
- *Sintaxe:*
FETCH [cursor] INTO [lista variáveis];
- *As variáveis devem ter sido declaradas*
- *Funciona como um SELECT ... INTO por linha da memória*

Exemplo: Fetch e Close

DECLARE

CURSOR C_CLIENTE IS

SELECT CD_CLIENTE, NM_CLIENTE FROM LOC_CLIENTE;

v_cliente c_cliente%rowtype; -- colunas do cursor

BEGIN

OPEN C_CLIENTE;

LOOP

FETCH C_CLIENTE INTO v_cliente;

EXIT WHEN C_CLIENTE%NOTFOUND;

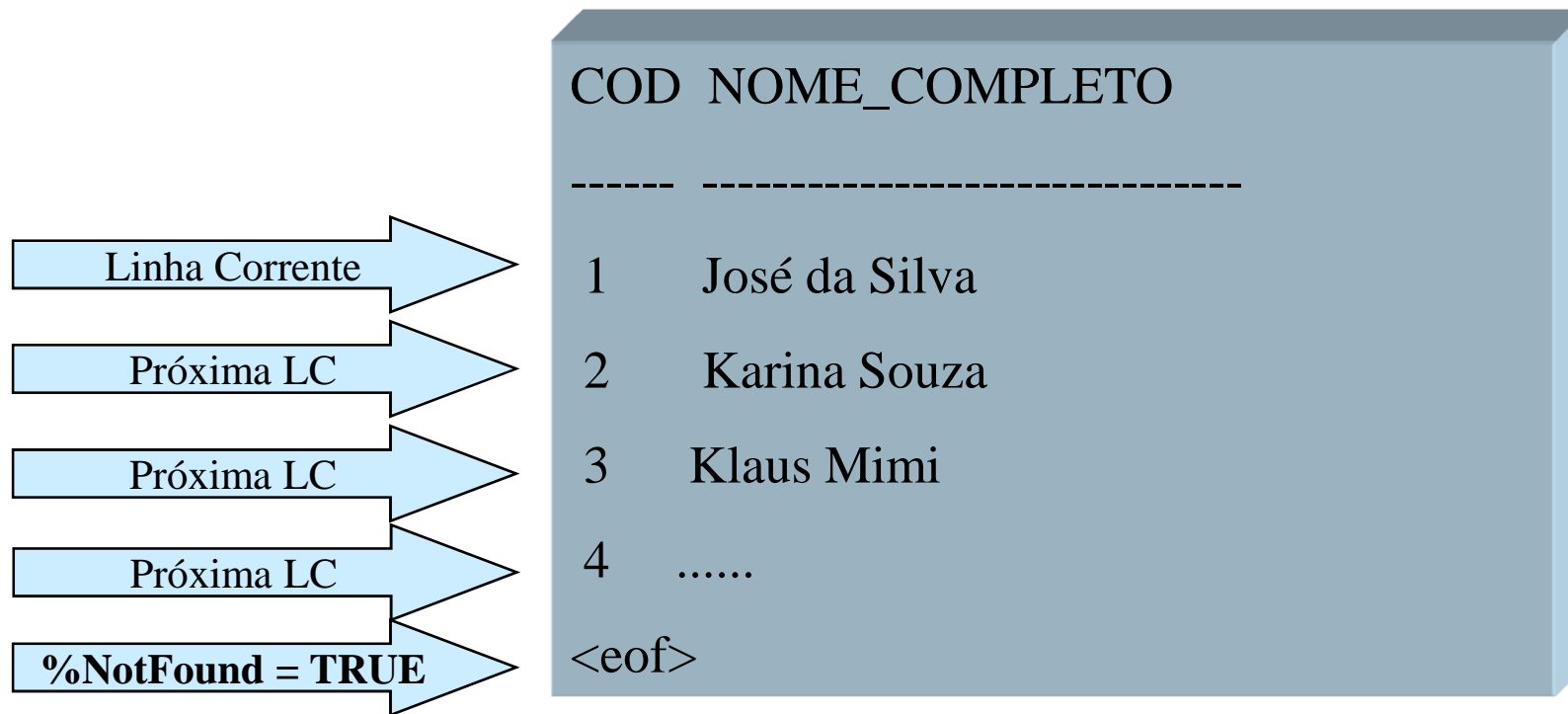
*DBMS_OUTPUT.PUT_LINE(RPAD(v_cliente.cd_cliente,7,'') || ' ' ||
RPAD(v_cliente.nm_cliente,50,' '));*

END LOOP;

CLOSE C_CLIENTE;

END;

- As linhas somente podem ser processadas quando seu conteúdo for transferido para variáveis. O fetch carrega para as variáveis e posiciona o cursor na próxima linha a ser lida. Quando chegar a última linha lida, ele ativa a variável interna %NOTFOUND = TRUE



Cursor Explícito: usando FOR...LOOP

- *O comando FOR...LOOP, quando for associado a um cursor, executa automaticamente, as seguintes ações :*
- *Cria a variável (tipo registro) que receberá os dados*
- *Abre o cursor*
- *Realiza a cópia das linhas, uma a uma (FETCH)*
- *Controla o final do cursor*
- *Fecha o mesmo*

Exemplo: cursor com FOR...LOOP

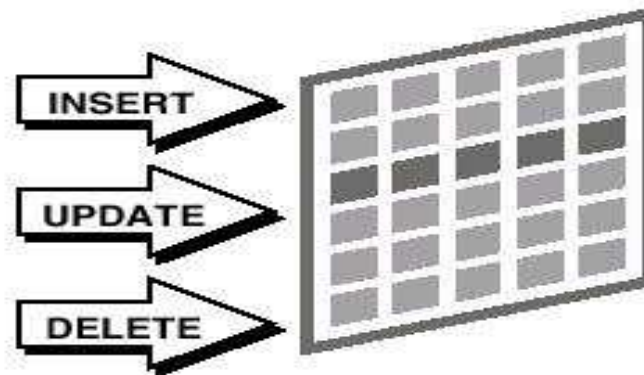
```
BEGIN
-- Subconsulta que é substituída internamente por um cursor pelo PL/SQL
FOR X IN (
SELECT CD_CLIENTE, NM_CLIENTE FROM LOC_CLIENTE
      )
LOOP
    DBMS_OUTPUT.PUT_LINE( RPAD(X.CD_CLIENTE,8,' ') || ' ' ||
        RPAD(X.NM_CLIENTE,60,' ') );
END LOOP;
END;
```

- Para executar transações **DML** (Insert, Update e Delete) em blocos PL/SQL, o Oracle abre implicitamente um cursor para processar o conjunto de dados resultante. Este cursor é chamado de cursor SQL
- Possíveis controles de execução de um cursor SQL
 - % FOUND => Encontrou informações processadas
 - % NOTFOUND => Não encontrou informações processadas
 - % ROWCOUNT => Quantas linhas foram processadas
 - % ISOPEN => O Cursor está aberto

Manipulando Dados Usando o PL/SQL

Alterar as tabelas de banco de dados usando os comandos DML:

- INSERT
- UPDATE
- DELETE



Atributos do Cursor SQL

Ao usar os atributos do cursor SQL, você poderá testar os resultados das instruções SQL.

SQL%ROWCOUNT	Número de linhas afetadas pela instrução SQL mais recente (um valor inteiro)
SQL%FOUND	Atributo booleano avaliado para TRUE se a instrução SQL mais recente afetar uma ou mais linhas
SQL%NOTFOUND	Atributo booleano avaliado para TRUE se a instrução SQL mais recente não afetar uma ou mais linhas
SQL%ISOPEN	Sempre é avaliado para FALSE porque o PL/SQL fecha os cursores implícitos imediatamente após a execução

Atributos do Cursor SQL

Deletar linhas que especificaram um número de ordem de compra a partir da tabela ITEM. Imprimir o número de linhas deletadas.

Exemplo

```
VARIABLE rows_deleted VARCHAR2(30)
DECLARE
  v_ordid  NUMBER := 605;
BEGIN
  DELETE FROM item
  WHERE      ordid = v_ordid;
  :rows_deleted := (SQL%ROWCOUNT ||
                    ' rows deleted.');
```

```
END;
/
PRINT rows_deleted
```

Exemplo: Cursor Implícito

```
DECLARE V_CD_GRUPO LOC_GRUPO.CD_GRUPO%TYPE := &COD_GRUPO;
BEGIN
    DELETE FROM LOC_GRUPO WHERE CD_GRUPO = V_CD_GRUPO;
    IF SQL%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('Não conseguimos encontrar o grupo de
veículos!');
    ELSE
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('Parabéns ! Exclusão realizada com sucesso. ');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE_APPLICATION_ERROR( -20002, 'Erro na exclusão=>' || SqlErrM );
END;
```

Exemplo: Cursor Implícito

```
BEGIN
  UPDATE LOC_CLIENTE SET NR_ESTRELAS = 5 WHERE CD_CLIENTE = 1;

  IF SQL%FOUND THEN
    DBMS_OUTPUT.PUT_LINE( 'TOTAL DE LINHAS PROCESSADAS=>' || SQL%ROWCOUNT );
  ELSE
    DBMS_OUTPUT.PUT_LINE( 'PROCESSAMENTO NÃO REALIZADO=>' || SQL%ROWCOUNT );
  END IF;
END;
```

Exemplo: Cursor com parâmetros

- *Usado para reaproveitamento do cursor a fim de alcançar melhor desempenho.*
- *O parâmetro é informado na cláusula OPEN.*
- *Também pode ser utilizado em cursores FOR...LOOP*

Exemplo: Exemplo Cursor com parâmetros

```
SET SERVEROUT PUT ON
DECLARE
V_NR_PLACA          LOC_VEICULO.NR_PLACA%TYPE;
V_NR_CHASSIS        LOC_VEICULO.NR_CHASSIS%TYPE;

CURSOR C_VEICULO( P_CD_PROPRIETARIO NUMBER, P_STATUS VARCHAR2) IS
SELECT NR_PLACA, NR_CHASSIS FROM LOC_VEICULO WHERE CD_PROPRIETARIO = P_CD_PROPRIETARIO AND STATUS
= P_STATUS;

BEGIN
    OPEN C_VEICULO(1, 'A');
    LOOP
        FETCH C_VEICULO INTO V_NR_PLACA, V_NR_CHASSIS;
        EXIT WHEN C_VEICULO%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE( V_NR_PLACA || ' ' || V_NR_CHASSIS);
    END LOOP;
    CLOSE C_VEICULO;
END;
```

Exemplo II: Exemplo Cursor com parâmetros

```
SET SERVEROUT PUT ON
DECLARE
CURSOR C_VEICULO( P_CD_PROPRIETARIO NUMBER, P_STATUS VARCHAR2) IS
SELECT NR_PLACA, NR_CHASSIS FROM LOC_VEICULO WHERE CD_PROPRIETARIO = P_CD_PROPRIETARIO AND STATUS
= P_STATUS;

BEGIN
  FOR REG_VEICULO IN C_VEICULO(1, 'A')
  LOOP
    DBMS_OUTPUT.PUT_LINE( REG_VEICULO.NR_PLACA || ' ' || REG_VEICULO.NR_CHASSIS);
  END LOOP;
END;
```

- *O que é e para que serve um Cursor*
- *Manipular o retorno de várias linhas em um comando SELECT*
- *Cursor implícito e explícito*
- *Cursor com parâmetros*

Não basta saber, é preciso exercitar.

Lista de Exercícios

Copyright © 2017

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, dos Professor (autor).